Proyecto Final

Inventario en Línea

Camargo Badillo Luis Mauricio
Gudiño Romero Miguel Angel

20 de mayo de 2025

Temas Selectos de Computación II
Desarrollo de APIs

David Bryan Padilla Alemán

Matemáticas Aplicadas y Computación

Índice

En la actualidad, la gestión eficiente de inventarios es un reto común para empresas y organizaciones de todos los tamaños. Muchos negocios aún dependen de métodos manuales o sistemas obsoletos, lo que genera errores, falta de visibilidad y dificultades para mantener información actualizada y confiable. La mejora de estos procesos es fundamental para mejorar la organización interna y reducir errores humanos.

Este proyecto propone una solución basada en una aplicación web full-stack que permite a los usuarios gestionar sus inventarios de manera eficiente y segura. La aplicación está diseñada para facilitar el registro, consulta, actualización y eliminación de productos en tiempo real, ofreciendo una herramienta moderna y accesible para la administración de inventarios.

Objetivo

El objetivo principal de este proyecto es diseñar y desarrollar una aplicación web que permita la gestión eficiente de inventarios en línea. Esta solución busca proporcionar a los usuarios una herramienta moderna, accesible y segura para administrar los productos de una tienda o empresa, facilitando el registro, consulta, actualización y eliminación de los artículos en tiempo real. El sistema estará orientado a mejorar la organización interna, reducir errores humanos y optimizar los procesos relacionados con el manejo de inventarios.

Problema por Resolver

Actualmente, muchas organizaciones enfrentan retos significativos en la administración de sus inventarios, especialmente cuando dependen de métodos manuales o sistemas poco actualizados. Entre los principales problemas se encuentran la falta de visibilidad sobre el estado actual del inventario, la dificultad para registrar y consultar movimientos de productos (altas, bajas, modificaciones y búsquedas), la posibilidad de cometer errores en el conteo o registro, y la ausencia de reportes claros y exportables que respalden las operaciones realizadas.

Propuesta de Solución

Para abordar el problema identificado, se desarrolló una aplicación web full-stack que integra tecnologías modernas tanto en el frontend como en el backend. El frontend fue implementado con React, permitiendo una interfaz de usuario dinámica, intuitiva y visualmente atractiva. La página principal presenta de manera clara las funcionalidades disponibles y cuenta con formularios de inicio de sesión y registro para garantizar la seguridad.

El backend se implementó utilizando Java Spring Boot, creando una API REST que permite la comunicación entre el frontend y la base de datos MySQL. Esta arquitectura facilita la escalabilidad y el mantenimiento del sistema, además de asegurar la integridad y persistencia de los datos.

Una vez autenticados, los usuarios pueden realizar todas las operaciones fundamentales del inventario: agregar nuevos productos, consultar el listado y detalles de los artículos, eliminar productos existentes o modificar la cantidad y características de los mismos. Cada operación se refleja en tiempo real en la interfaz y el sistema proporciona la opción de descargar un reporte en PDF con los resultados de la operación realizada. Así, se garantiza que los usuarios

cuenten con información actualizada y respaldada, mejorando la eficiencia y confiabilidad en la gestión de inventarios.

Diseño de la Solución

Este sistema de gestión de inventarios fue diseñado bajo una arquitectura web full-stack, separando claramente la interfaz de usuario (frontend), la API (backend), y la persistencia de datos (base de datos).

Componentes

El sistema se compone de tres componentes principales:

- **Frontend**: Aplicación desarrollada en React, responsable de la interacción con el usuario, validación de formularios y consumo de la API REST.
- **Backend**: API REST construida con Java Spring Boot, encargada de la lógica, autenticación, autorización y comunicación con la base de datos.
- **Base de Datos**: Servidor MySQL que almacena la información de usuarios, productos, inventario y operaciones.

Interacción entre Componentes

- El usuario accede a la aplicación web desde su navegador y realiza acciones como iniciar sesión, registrar productos o consultar el inventario.
- El frontend envía solicitudes HTTP a la API REST del backend, que valida y procesa las peticiones.
- El backend interactúa con la base de datos MySQL para almacenar o recuperar información según la operación solicitada.
- Las respuestas del backend son procesadas por el frontend, que actualiza la interfaz de usuario en tiempo real.
- Para ciertas operaciones, el usuario puede exportar la información visualizada en la interfaz a un archivo PDF.

Casos de Uso

El sistema tiene los siguientes casos de uso principales:

- Registro y autenticación de usuarios.
- Alta, baja, modificación y consulta de productos en el inventario.
- Exportación de reportes en PDF.

El sistema implementa autenticación basada en JWT (JSON Web Tokens) para proteger los endpoints sensibles y garantizar que solo los usuarios autorizados puedan realizar operaciones sobre el inventario.

A continuación, los diagramas para los distintos casos de uso:

Modelo de Datos

La base de datos relacional está compuesta por las siguientes tablas principales:

- **users**: Almacena la información de los usuarios registrados, incluyendo credenciales.
- **items**: Contiene los datos de los productos disponibles, incluyendo nombre, descripción, precio y cantidad.
- **operations**: Guarda un historial de las operaciones realizadas sobre el inventario.

Tecnologías Utilizadas

El desarrollo del sistema requirió la integración de diversas tecnologías modernas que se seleccionaron en función a su facilidad de uso y compatibilidad con los objetivos del proyecto.

- **React**: Biblioteca de JavaScript utilizada para construir la interfaz de usuario (frontend). Permite crear componentes reutilizables y una experiencia dinámica e interactiva para el usuario.
- **Vite**: Herramienta de construcción y desarrollo rápido para proyectos frontend en React, utilizada para optimizar el flujo de trabajo y la velocidad de desarrollo.
- **Mantine UI**: Biblioteca de componentes de interfaz para React, empleada para lograr una apariencia moderna en la aplicación, sin tener que desarrollar los componentes desde cero.
- **Java Spring Boot**: Framework para el desarrollo del backend, encargado de la exposición de la API REST y la gestión de la seguridad y autenticación.
- **MySQL**: Sistema de gestión de bases de datos relacional, utilizado para almacenar de manera persistente la información de usuarios, productos y operaciones.
- **JWT (JSON Web Tokens)**: Tecnología de autenticación y autorización, utilizada para proteger los endpoints sensibles.
- **Docker**: Plataforma de contenedores empleada para facilitar el despliegue y la ejecución del sistema en cualquier entorno, asegurando la portabilidad.
- **Docker Compose**: Herramienta para orquestar múltiples contenedores de Docker (frontend, backend y base de datos) y simplificar su despliegue.
- **jsPDF**: Librería de JavaScript utilizada en el frontend para exportar los reportes de inventario en formato PDF.
- **Axios**: Cliente HTTP para JavaScript, empleado en el frontend para realizar solicitudes a la API REST del backend.

Ejecución del Proyecto con Docker

Para facilitar la ejecución y despliegue del sistema, se utilizó Docker. Esto permite levantar todos los servicios necesarios (frontend, backend y base de datos) con un solo comando, sin preocuparse por dependencias o configuraciones específicas del entorno local.

Requisitos Previos

- Tener instalado Docker y Docker Compose en el sistema.
- No tener otros servicios ocupando los puertos 80, 8080 o 3306.

Instrucciones

Desde la raíz del proyecto, ejecutar:

```
docker-compose up --build
```

Esto construirá y levantará todos los servicios. La aplicación frontend estará disponible en `http://localhost`, el backend en `http://localhost:8080` y el servidor MySQL en el puerto 3306 (`localhost:3306`).

Casos de Ejecución

(falta)

Conclusión

A lo largo del proyecto, se logró diseñar e implementar una solución web full-stack funcional, segura y escalable, integrando herramientas modernas tanto en el frontend como en el backend.

Entre los principales aprendizajes que se obtuvieron durante el desarrollo de este proyecto destacan la importancia de la planificación arquitectónica (no se puede crear un buen proyecto sin antes planificarlo), la correcta separación de responsabilidades entre los distintos componentes del sistema (¿qué es trabajo del backend y qué del frontend?) y la adopción de buenas prácticas de desarrollo, como el uso de control de versiones (se usó git y GitHub, aprovechando especialmente la funcionalidad de las ramas) y contenedores.

Como áreas de mejora y expansión futura para este proyecto, se identifican:

- Implementar funcionalidades avanzadas, como notificaciones automáticas de bajo inventario o integración con sistemas externos de facturación.
- Mejorar la interfaz de usuario para dispositivos móviles.
- Incorporar pruebas automatizadas.
- Permitir la gestión de múltiples almacenes o sucursales.
- Optimizar el rendimiento y la seguridad ante escenarios de mayor concurrencia.

Estas mejoras no solo enriquecerían la funcionalidad del sistema, sino que también ofrecerían una experiencia de usuario más completa y satisfactoria.

No obstante, no hay que olvidar que el sistema desarrollado cumple con los objetivos planteados: permite a los usuarios gestionar inventarios en línea, realizar operaciones fundamentales sobre los productos y exportar reportes en PDF. Además, la arquitectura modular facilita el mantenimiento y la futura expansión del sistema.